

Задача А. Запросы Таро на дереве

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	10 секунд
Ограничение по памяти:	256 мегабайт

У кота Таро есть подвешенное дерево из n вершин. Вершины пронумерованы от 0 до $n - 1$, включительно. Вершина с номером 0 является корнем дерева. На каждом ребре написан целый положительный вес.

Таро хочет выполнить m запросов. Каждый запрос состоит из двух вершин u и v . Запросы следует выполнить следующим образом:

- Если u равно v , выведите -1 .
- Иначе, если вершина v лежит в поддереве вершины u , выведите максимальный вес на ребрах на простом пути из v в u .
- Иначе, удалите ребро, соединяющее вершину u и ее родителя. Вместо него добавьте новое ребро (с таким же весом), которое сделает вершину u ребенком вершины v . (Обратите внимание, что все поддерево вершины u теперь часть поддерева вершины v .) Ничего выводить не нужно.

Формат входных данных

В первой строке содержатся два целых числа n и m ($1 \leq n, m \leq 200\,000$) — количество вершин и запросов, соответственно.

Во второй строке содержится $n - 1$ целое число p_i ($1 \leq i \leq n - 1$, $0 \leq p_i < i$) — родитель i -й вершины.

В третьей строке содержится $n - 1$ целое число w_i ($1 \leq i \leq n - 1$, $0 \leq w_i \leq 1\,000\,003$) — вес ребра из вершины i в ее родителя.

В следующих m строках содержится по два целых числа u_i и v_i ($0 \leq v_i, u_i \leq n - 1$) — запросы.

Формат выходных данных

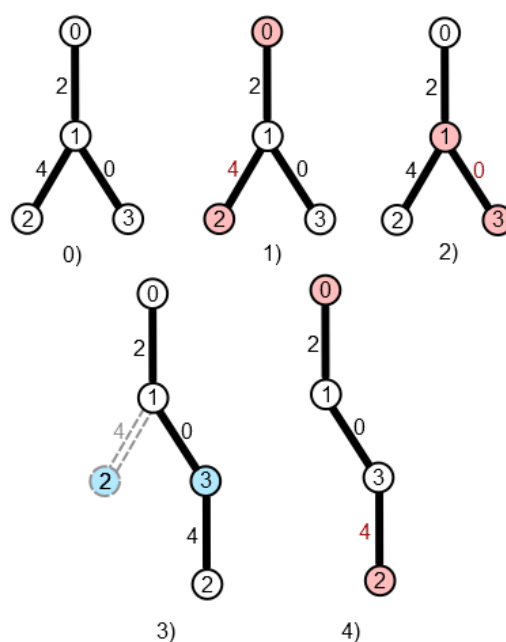
Выведите ответы на запросы.

Примеры

стандартный ввод	стандартный вывод
4 4 0 1 1 2 4 0 0 2 1 3 2 3 0 2	4 0 4
7 10 0 1 1 2 4 4 5 4 0 2 4 6 0 1 1 1 2 6 2 4 2 5 0 2 0 2 1 5 1 5 2 6	5 -1 6 2 4 5 5 4 4 6
10 4 0 0 1 2 3 5 6 7 8 14 24 25 65 13 59 19 37 58 1 9 5 6 2 4 6 8	59 59 65 37

Замечание

Пояснение к первому примеру.



Задача В. Прямоугольники

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Когда-то тут была легенда про вёдра, но её съели.

Жюри олимпиады

Есть таблица T размера $N \times M$. Элементами таблицы являются прямоугольники T_{ij} , где $0 \leq i < N$ и $0 \leq j < M$. Прямоугольник T_{ij} задаётся четвёркой чисел $(x_1^{ij}, y_1^{ij}, x_2^{ij}, y_2^{ij})$, где (x_1^{ij}, y_1^{ij}) и (x_2^{ij}, y_2^{ij}) — координаты противоположных углов прямоугольника. Стороны прямоугольника параллельны осям координат.

Далее вам поступают запросы. Каждый запрос состоит из четырёх чисел: (r_1, c_1, r_2, c_2) . Ответом на такой запрос является площадь фигуры, являющейся пересечением всех прямоугольников T_{ij} таких, что $\min(r_1, r_2) \leq i \leq \max(r_1, r_2)$ и $\min(c_1, c_2) \leq j \leq \max(c_1, c_2)$. Запросов очень много, поэтому мы просим вас вывести сумму ответов на все запросы по модулю $10^9 + 7$.

Формат входных данных

В первой строке записаны два целых числа N и M — размеры таблицы T ($1 \leq N, M \leq 127$). Далее в N строках описывается таблица T : в $(i+1)$ -й строке $(j+1)$ -я четвёрка чисел $x_1^{ij} y_1^{ij} x_2^{ij} y_2^{ij}$ описывает прямоугольник T_{ij} . Гарантируется, что $|x_k^{ij}|, |y_k^{ij}| \leq 10^6$.

Дальше в отдельной строке записано четыре числа. Первое из них, число Q — количество запросов ($1 \leq Q \leq 5 \cdot 10^6$). Следующие три числа — это A, B, v_0 ($0 \leq A, B, v_0 < 10^9 + 7$). При помощи этих чисел генерируется бесконечная последовательность $\{v_i\}$ по правилу $v_i = (A \cdot v_{i-1} + B) \bmod (10^9 + 7)$.

После этого k -й запрос (запросы нумеруются с единицы) задаётся следующей четвёркой чисел: $(v_{4k-3} \bmod N, v_{4k-2} \bmod M, v_{4k-1} \bmod N, v_{4k} \bmod M)$.

Формат выходных данных

Выведите сумму ответов на все запросы по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
2 2 0 0 2 2 1 1 3 3 0 3 2 1 1 2 3 0 1 500000003 4 2	1
3 2 8 -1 -7 6 6 8 9 10 -4 -10 4 9 -3 -8 6 9 -2 -9 3 8 -5 7 7 3 5 303164476 273973578 65779139	85

Замечание

В первом примере запрос имеет вид $(1, 0, 0, 1)$, то есть это запрос ко всей таблице. Пересечением всех прямоугольников является квадрат с углами в точках $(1, 1)$ и $(2, 2)$. Его площадь равна 1.

Во втором примере запросы имеют вид $(0, 1, 1, 1)$, $(1, 0, 2, 0)$, $(0, 0, 2, 1)$, $(0, 1, 1, 1)$, $(0, 1, 0, 0)$. На второй запрос ответ — 85, на остальные — 0.

Задача С. Двумерные запросы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам задан массив размера 2^{17} . Требуется ответить на запросы: сколько есть элементов $f[i]$ таких, что $l \leq i \leq r$ и $x \leq f[i] \leq y$.

Формат входных данных

На первой строке число q ($1 \leq q \leq 2^{17}$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
0. unsigned int a, b; // даны во входных данных
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand17() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur » 15; // число от 0 до  $2^{17} - 1$ .
5. }
6. unsigned int nextRand24() {
7.     cur = cur * a + b; // вычисляется с переполнениями
8.     return cur » 8; // число от 0 до  $2^{24} - 1$ .
9. }
```

Сначала массив генерируется следующим образом:

```
1. for (int i = 0; i < 1 « 17; i++)
2.     f[i] = nextRand24();
```

Потом генерируются запросы следующим образом:

```
1. l = nextRand17();
2. r = nextRand17();
3. if (l > r) swap(l, r); // получили отрезок [l..r]
4. x = nextRand24();
5. y = nextRand24();
6. if (x > y) swap(x, y); // получили отрезок [x..y]
7. b += c; // c -- ответ на данный запрос, для ответа на запросы в online
```

Формат выходных данных

Выведите сумму ответов на все запросы второго типа по модулю 2^{32} .

Примеры

стандартный ввод	стандартный вывод
5 13 239	111139

Задача D. Жадность

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В этой задаче вам придется решить одну из разновидностей известной задачи о рюкзаке. К сожалению, она не NP-полная.

У вас есть n предметов, каждый из которых имеет размер s_i и стоимость c_i . Также у вас q рюкзаков, каждый из которых описывается своим размером x_i . Рюкзак размера x может содержать набор предметов суммарным размером не более x .

Чтобы заполнить рюкзак, используется следующий алгоритм:

1. Сортируются все предметы по невозрастанию их размеров.
2. Предметы просматриваются в полученном порядке.
3. Предмет кладётся в рюкзак, если в рюкзаке достаточно свободного места.

Для каждого из q рюкзаков выведите суммарную стоимость предметов в этом рюкзаке после запуска алгоритма, описанного выше.

Формат входных данных

Первая строка входного файла содержит одно целое число n : число предметов ($1 \leq n \leq 10^5$).

Вторая строка содержит n целых чисел s_1, s_2, \dots, s_n : размеры предметов ($1 \leq s_i \leq 10^6$).

Третья строка содержит n целых чисел c_1, c_2, \dots, c_n : стоимости предметов ($1 \leq c_i \leq 10^4$).

Четвертая строка содержит одно целое число q : количество рюкзаков ($1 \leq q \leq 10^5$).

Пятая строка содержит q целых чисел x_1, x_2, \dots, x_q : вместимости рюкзаков ($1 \leq x_i \leq 10^6$).

Предметы даны в порядке невозрастания их размеров. Вы не должны менять их порядок.

Формат выходных данных

Для каждого рюкзака выведите суммарную стоимость предметов внутри него в отдельной строке.

Решать задачу для каждого рюкзака вы должны независимо от других.

Пример

стандартный ввод	стандартный вывод
5	1
5 4 4 2 1	70
10 1 100 20 40	171
3	
4 8 100	

Задача Е. Бенни и сумма

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вероятно, вы уже поняли, что Бенни очень любит различные типы запросов.

Иногда, когда она идет гулять с друзьями, они идут в кино. Но каждый раз, когда она идет куда-то, она не может расслабиться, если ее домашняя работа еще не выполнена.

Она любезно просит вас снова помочь ей с домашним заданием.

В этой задаче вам дан массив p , состоящий из n элементов.

Вам нужно ответить на m запросов.

Каждый запрос имеет следующий формат: $l \ r \ x$. Ответ на запрос — $\sum_{i=l}^r (p_i \bmod x)$.

Для каждого запроса выведите ответ на отдельной строке.

Формат входных данных

В первой строке содержатся два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$).

Во второй строке содержится n целых чисел p_i ($0 \leq p_i \leq 2 \cdot 10^5$).

В следующих m строках содержится по три целых числа: l , r и x ($1 \leq l \leq r \leq n$, $1 \leq x \leq 2 \cdot 10^5$).

Формат выходных данных

Для каждого запроса выведите ответ на отдельной строке.

Пример

стандартный ввод	стандартный вывод
10 5	11
23 32 42 50 2 33 41 5 100 3	75
2 10 3	36
1 9 23	9
3 10 12	13
2 8 4	
2 10 5	

Замечание

- $2 + 0 + 2 + 2 + 0 + 2 + 2 + 1 + 0 = 11$
- $0 + 9 + 19 + 4 + 2 + 10 + 18 + 5 + 8 = 75$
- $6 + 2 + 2 + 9 + 5 + 5 + 4 + 3 = 36$
- $0 + 2 + 2 + 2 + 1 + 1 + 1 = 9$
- $2 + 2 + 0 + 2 + 3 + 1 + 0 + 0 + 3 = 13$

Задача F. Visits

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	12 секунд
Ограничение по памяти:	256 мегабайт

Сергей — замечательный человек, за последние 21 год он был почтальоном, банкиром, конькобежцем, и даже королем! Неудивительно, что у него много друзей. Но разносторонние профессии и различные места работы, к сожалению, заставили его потерять связь с большинством старых друзей... Настало время это изменить! Сергей начинает грандиозный тур по Сергеотии для восстановления дружеских связей.

В Сергеотии n городов, соединенных в сеть двухсторонними дорогами ($n - 1$ дорог). Наш герой хочет посетить каждый город и уже установил для себя порядок их посещения. Из каждого города к другому он будет перемещаться на автомобиле, арендованном у компании SRC (Sergeoti Rental Company). Каждая аренда является абсолютно бесплатной, но автомобили нуждаются в заправке — автомобиль с емкостью бака k должен заправляться в самом начале, а затем снова после каждых k пройденных дорог.

SRC хорошо знает плана Сергея и тот факт, что он намерен завершить свое путешествие как можно скорее. Поэтому они ловко выбрали объемы баков арендованных автомобилей, что в каждом городе Сергей будет (пере)заправлять две машины: прибывающую и уезжающую (за исключением начального и конечного городов, где лишь только по одному автомобилю, нуждающихся в заправке.).

Зная маршрут Сергея, цены на топливо и объемы баков, определить стоимость каждого сегмента маршрута.

Формат входных данных

Первая строка входного файла содержит целое число n ($2 \leq n \leq 50\,000$) — количество городов в Сергеотии. Города пронумерованы от 1 до n . На следующей строке находится последовательность из n целых чисел c_1, \dots, c_n ($1 \leq c_i \leq 10\,000$), разделенных пробелами, которая задает цены топлива для любого автомобиля в городах Сергеотии: c_i — стоимость заправки топливом в городе с номером i . (Заправиться вы можете только, отдав c_i и никак не меньше).

Последующие $n - 1$ строк задают дорожную сеть в Сергеотии. Каждая строка содержит по два целых числа a и b ($1 \leq a, b \leq n$), которые указывают на существование двухсторонней дороги между городами a и b . После этого содержится строка с последовательностью из n целых чисел t_1, \dots, t_n , которые определяют порядок, в котором Сергей собирается посещать города (гарантируется, что каждый город в последовательности появляется ровно по одному разу).

И наконец, последняя строка содержит последовательность из $n - 1$ целых чисел k_1, \dots, k_{n-1} , описывающих объемы баков арендованных машин: k_i соответствует поездке из города t_i в город t_{i+1} . Сергей должен перезаправить свою машину после каждых k_i пройденных дорог. Гарантируется, что расстояние между этими городами всегда делится на k_i .

Формат выходных данных

Выведите $n - 1$ строк, в каждой строке по одному целому числу. Число в i -й строке должно соответствовать суммарной стоимости перезаправки на пути из города t_i в город t_{i+1} .

Пример

стандартный ввод	стандартный вывод
5	10
1 2 3 4 5	6
1 2	10
2 3	5
3 4	
3 5	
4 1 5 2 3	
1 3 1 1	